# Embedded Systems

# Assignment 1

# The Vault

For this practical you will need the following:

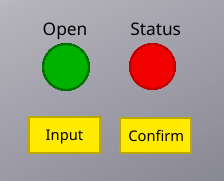
1. Arduino Uno

2. Open Smart Rich Shield

In this assignment you’ll combine your knowledge of LEDs and buttons. You will make the controls for a safe much like a safe in a hotel room or a digital safety locker in a swimming pool. The version you’ll build will be a simplified version. It does not have a display and only uses two buttons and two LEDs.

The vault is secured by an entry code. It consists of 4 digits. Each digit has the value 1-4. Example of a valid entry code: '4 3 1 2'. To enter a code and to check the input, the vault is equipped with 2 buttons, a red LED and a green LED.

The input panel on the door of the Vault could look something like this:



We will simulate this input panel with the buttons and LEDs on the Shield.

### User interaction description

In the resting state no LEDs are on. When the Vault is unlocked the green LED is on.

To enter the secret code the user takes the following steps.

1. Push the input button: both red and green light up for 1 second and switch off. A new input sequence starts.
2. The user pushes the input button to enter the first digit 1 click = 1, 2 clicks = 2, etc. Each time the input button is pushed the green led lights up once. When the user is done with the digit, they click “confirm.”, the red light lights up briefly to signal the user they can input the next digit. Repeat step 2 until 4 digits have been entered.
3. When the user has entered the four digits, the safe either switches on the green light to indicate the safe is open, or switches the red led on for 2 seconds to indicate a wrong code.
4. the process can start over again at 1.

### Approach

To get started we’ll divide this problem into sub-problems.

1. First, we’ll attempt to verify only a single digit (1-4) If the digit we enter is equal to the secret code, the vault opens.
2. Make a button light up the green LED for 500ms.
3. Make sure your code registers the button only once.
4. Increase a counter when the button is pushed. (Show the counter value in the serial monitor to check if it works)
5. Make sure that the counter is limited from 1-4
6. Add the second button (confirm button)
7. When the confirm button is pushed, compare the current counter to the secret code.
8. If correct, light up the green LED, otherwise the red LED.

Did you choose good variables names?

Did you add good comments?

1. Now we’ll attempt to add another digit.
2. Complete assignment part A step 1-5.
3. Add a new counter variable to keep track of the second digit.
4. Add a new variable, named digit\_pos, that indicates which digit the user is entering.
5. From step 5, when the confirm button is pushed, change the digit\_pos variable so that you can keep track which digit the user is entering.
6. Now, when the input button is pushed, increase the second counter and not the first.
7. When the user presses the confirm button, compare the code to the secret.

Let say that the user entered 4 and 2 as the two digits. Now you have two variables d1 == 4 and d2 == 2. Think about how you can take the two separate variables and compare them to a single variable with the secret value.

1. Compare the combined d1 & d2 to the secret.
2. If the result is correct, light up the green LED, otherwise the red LED.
3. Add the 3rd and 4th digit.

You could do this by just replicating what you did before. This is fine.

1. Improve the code.

Can you make it work by reducing variables and duplicated code?

Extra: devise a way to let the user program their personal secret code using the input panel. Show your flow chart to the teacher and make changes to the program.

### Formative indications

After hand-in you will be given feedback in this assignment using the following criteria.

1. General
   1. Variable names
   2. Code indentation
   3. Comments.
2. Completion of Part A

The next parts are optional but will improve your formative indication.

Completion of Part B, C, D

All behaviour specified in the user interaction description.

Flow chart of the code that you hand in.